

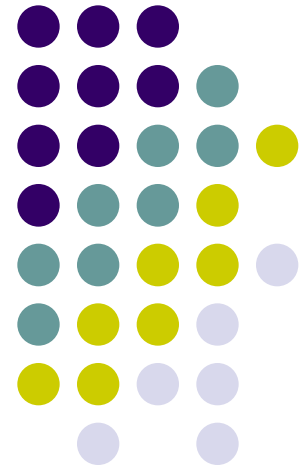
CS 460

Programming Languages

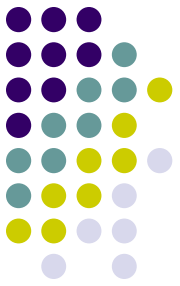
Fall 2023

Dr. Watts

(11 September 2023)

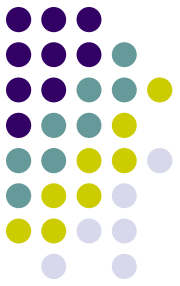


Course Administration



- Exercise 1 Redo
- Exercise 2 Preliminary Exercise
- Project 1 Prelim will be assigned on Wednesday

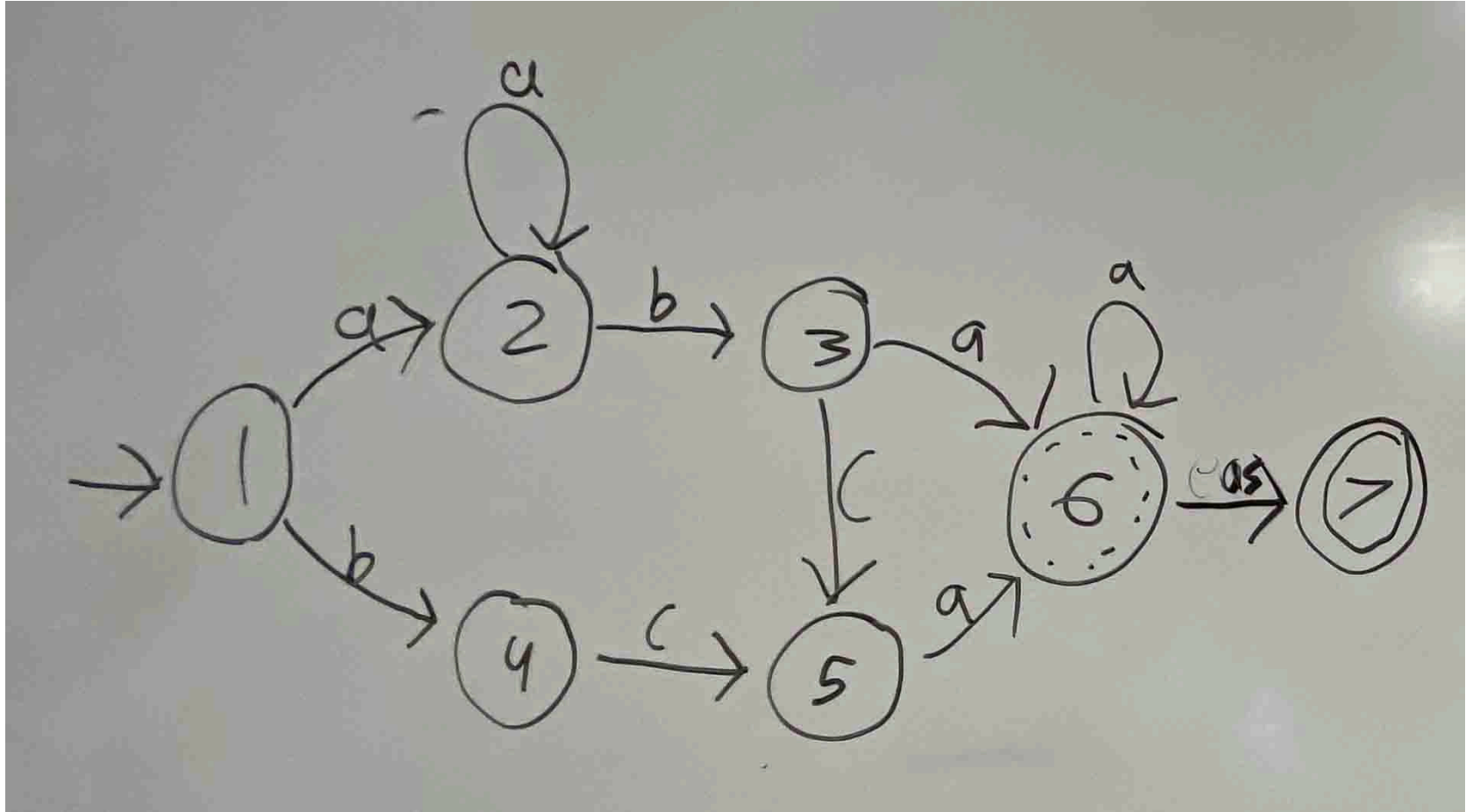
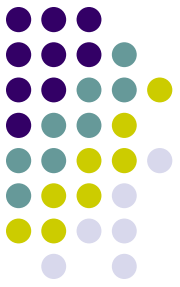
DFAs as scanners (aka tokenizers)



- Alphabet = {a, b, c, x, y, z, -}
- Regular expression 1 (RE1)
 - $a^* (ab \mid bc) a^+$
 - DFA
- Regular expression 2 (RE2)
 - $x^+ (xy \mid yz \mid xz) z^*$
 - DFA

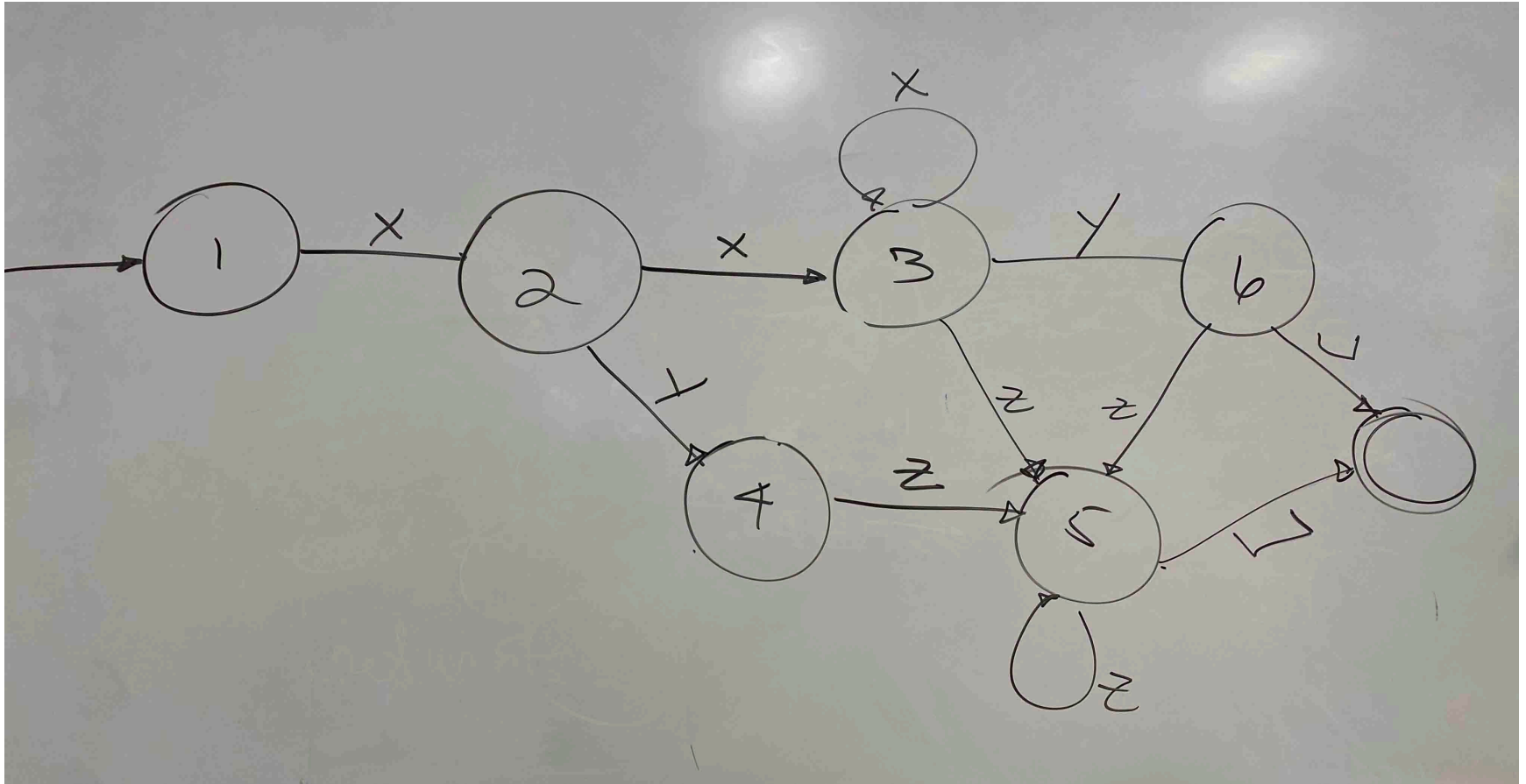
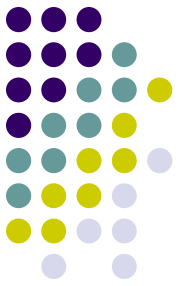
Regular expression 1 (RE1)

$a^* (ab \mid bc) a^+$



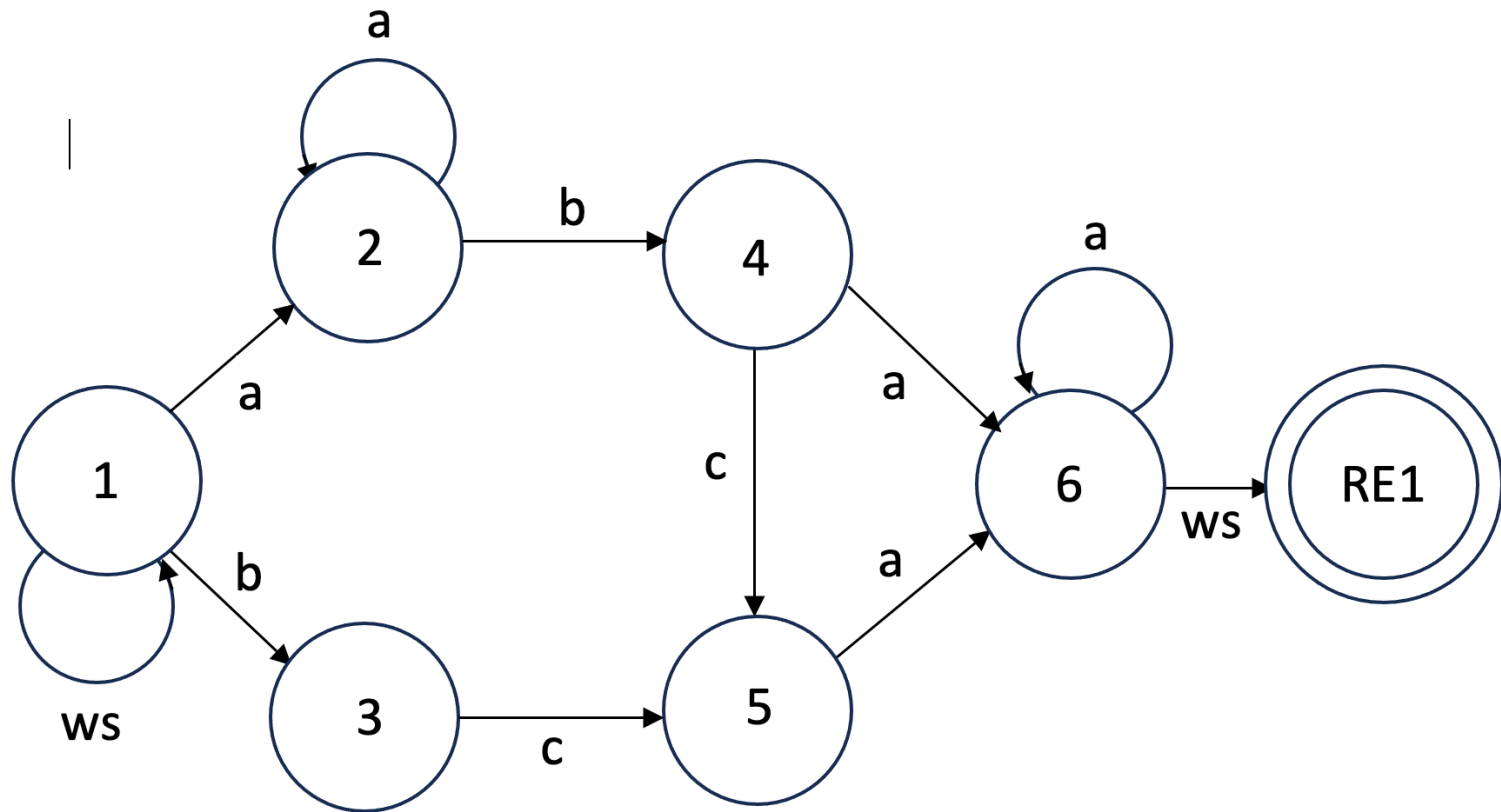
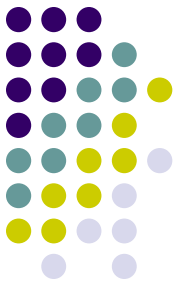
Regular expression 2 (RE2)

$x^+ (xy \mid yz \mid xz) z^*$



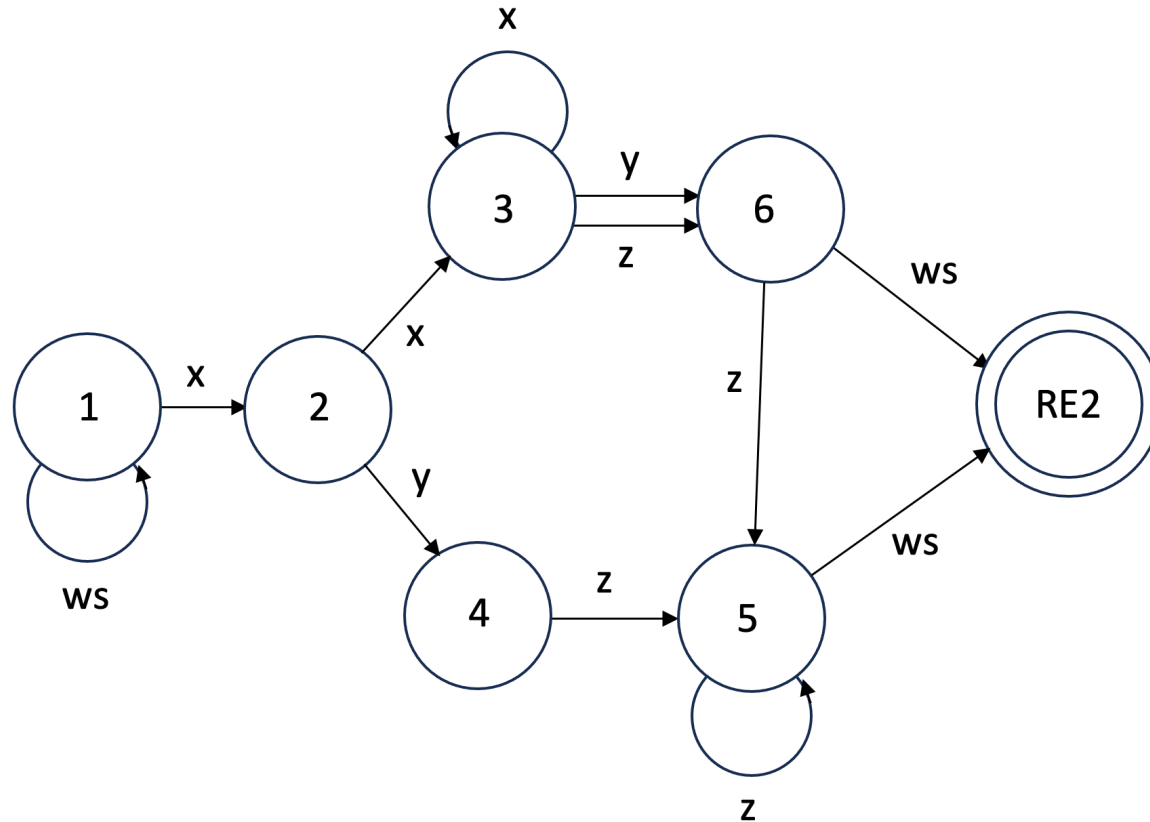
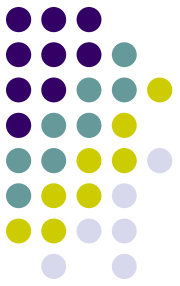
Regular expression 1 (RE1)

$a^* (ab \mid bc) a^+$



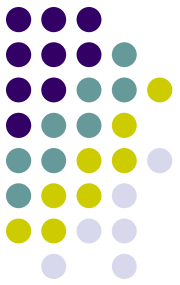
Regular expression 2 (RE2)

$x^+ (xy \mid yz \mid xz) z^*$



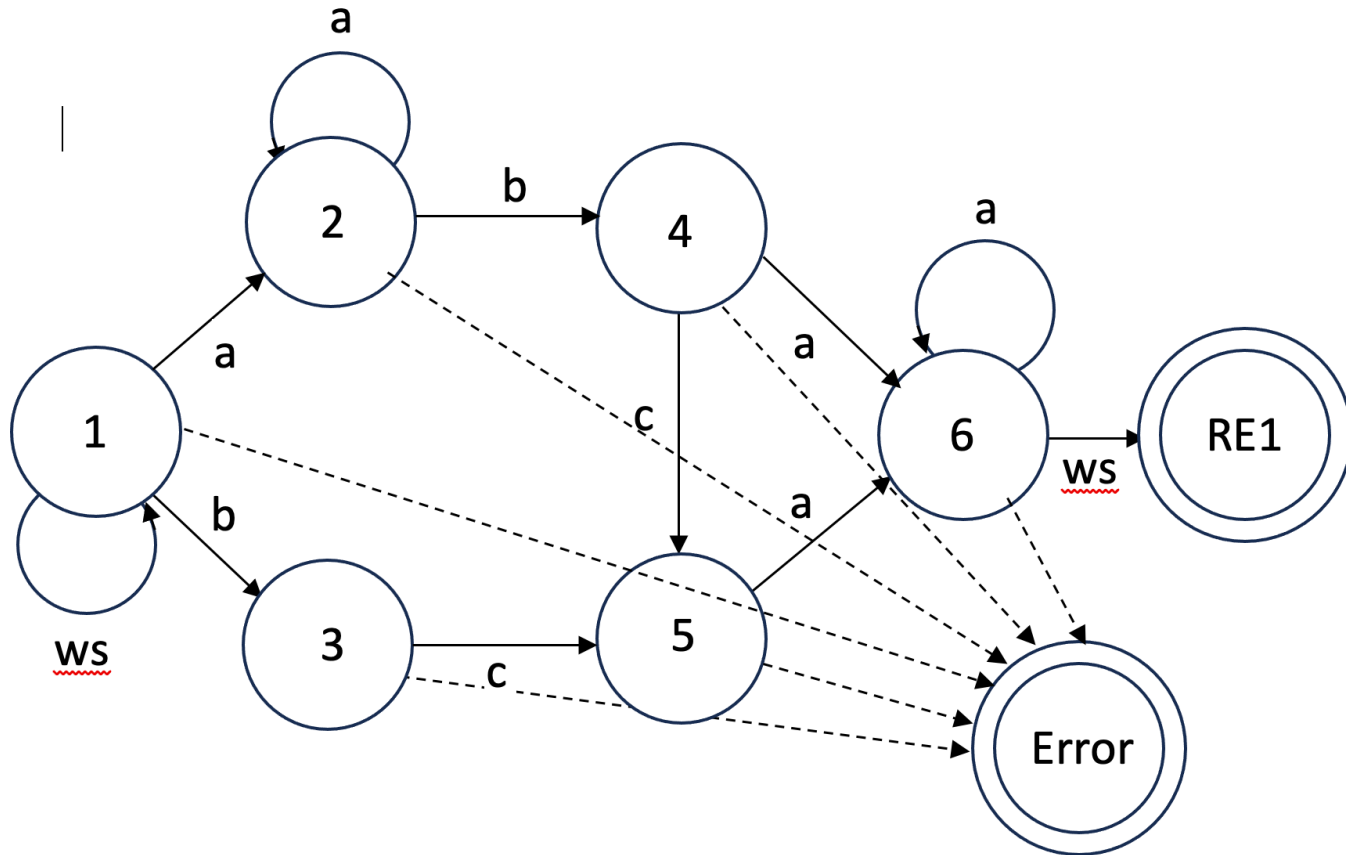
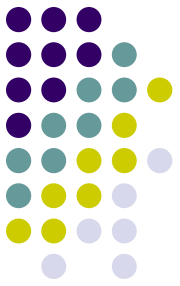
Adding terminating states

- Successful states
- Error states



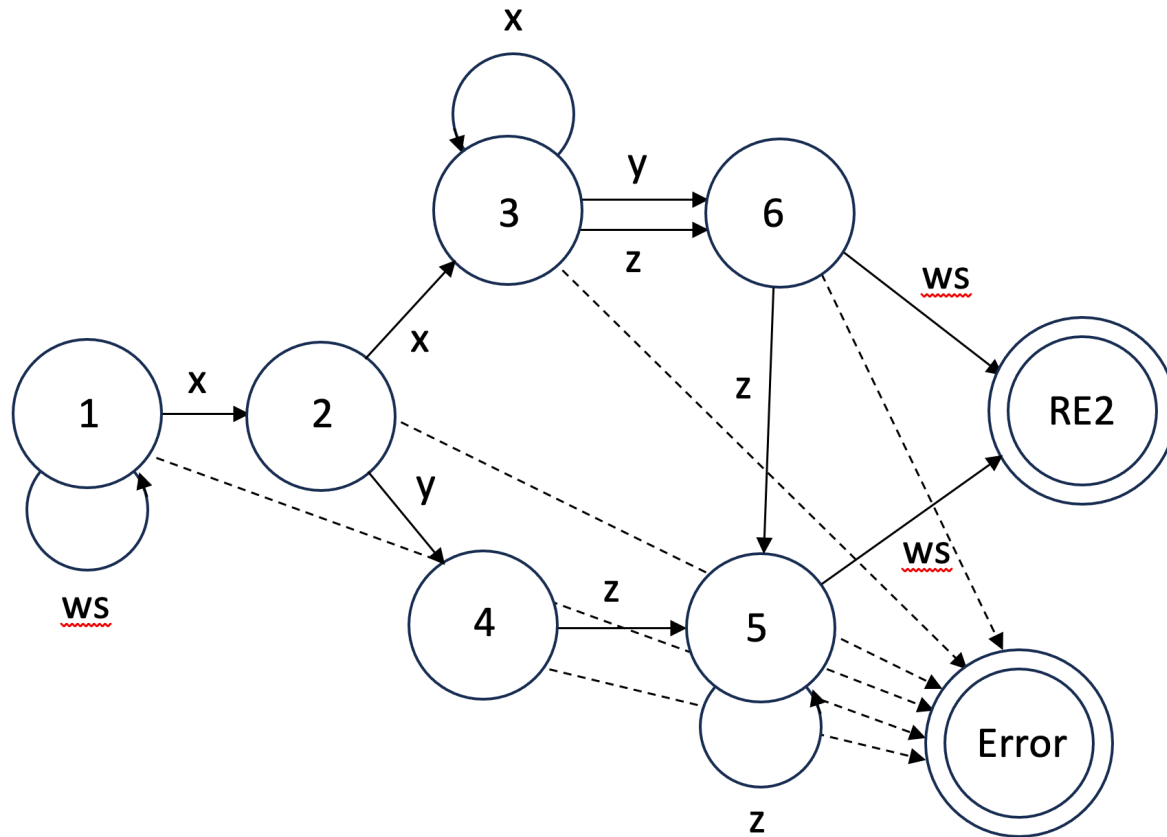
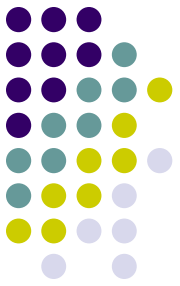
Regular expression 1 (RE1)

$a^* (ab \mid bc) a^+$

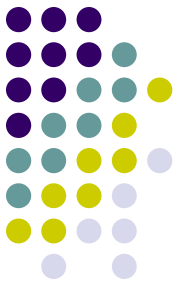


Regular expression 2 (RE2)

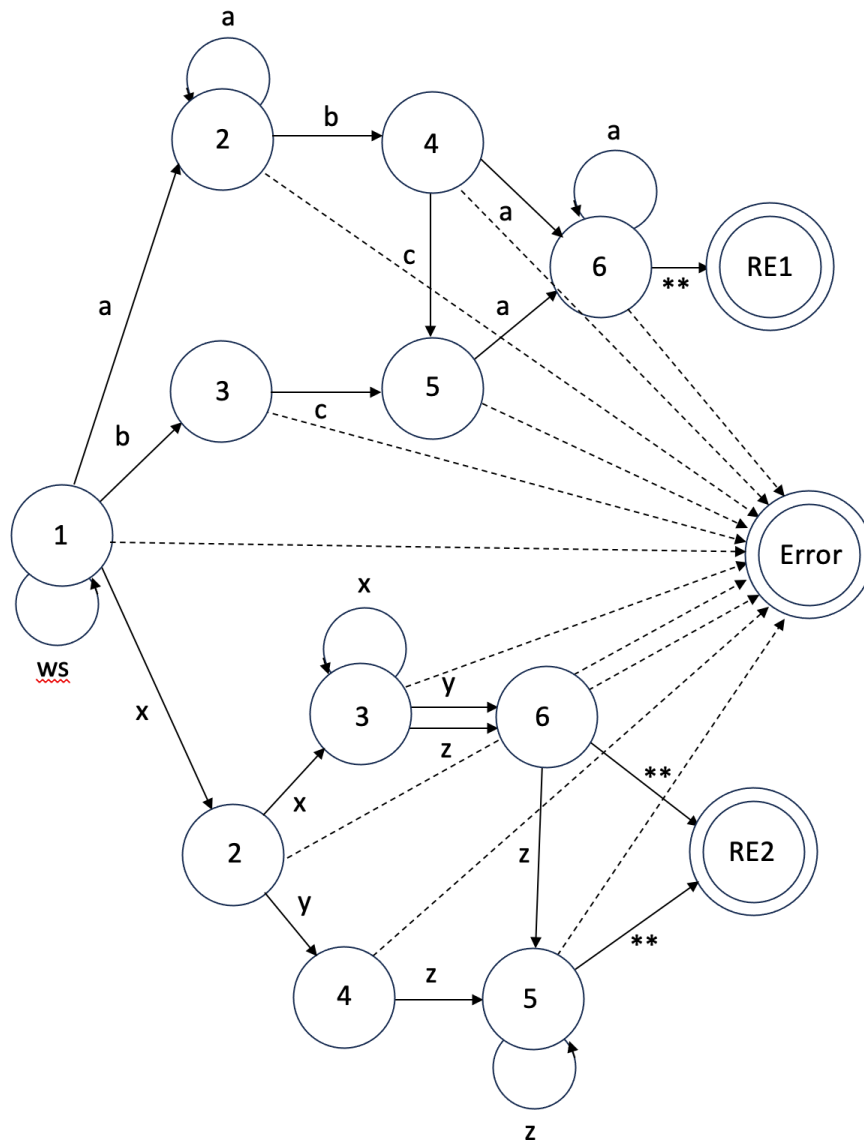
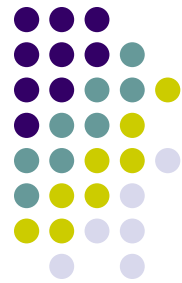
$x^+ (xy \mid yz \mid xz) z^*$

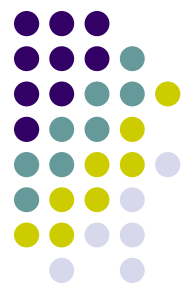
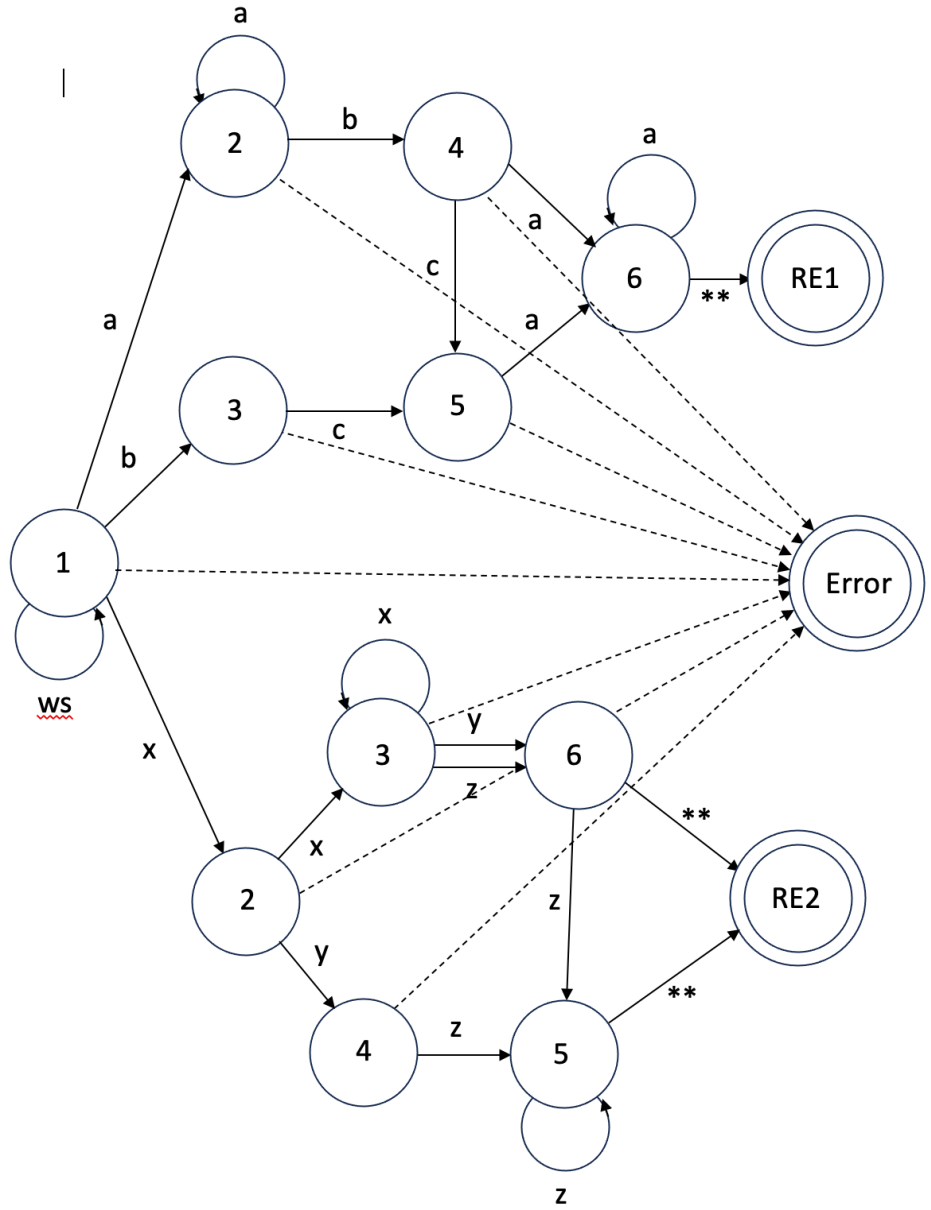


Combined



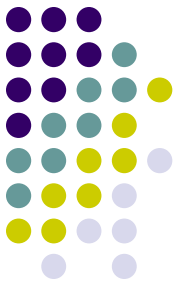
- $(a^* (ab \mid bc) a^+) \mid (x^+ (xy \mid yz \mid xz) z^*)$



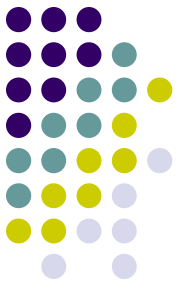


Programming a DFA

- Table

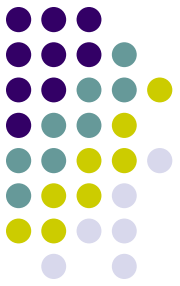


Regular Expression for Numeric Literals

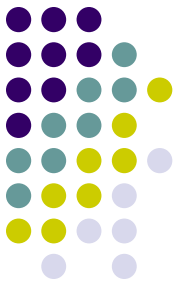


- Regular expression for general class of numeric literals signed/unsigned and integer/real
- Alphabet = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, -, +, .}
- Regular Expression
- How do you recognize the end of a numeric literal?

Regular Expression for Numeric Literals



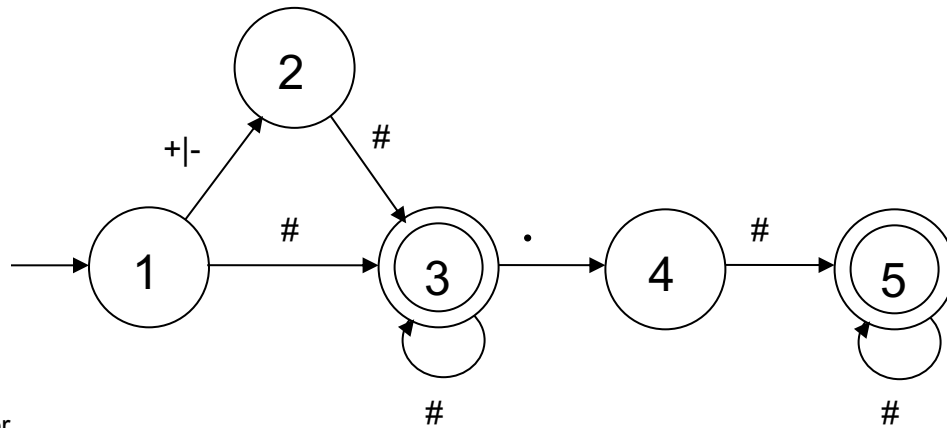
$(+|-|\lambda)(0|1|2|3|4|5|6|7|8|9)+(\.(0|1|2|3|4|5|6|7|8|9)+|\lambda)$
 $(\text{plus}|\text{minus}|\lambda) (\#+ (\cdot|\lambda) \#^*) | (\#^* (\cdot|\lambda) \#+)$

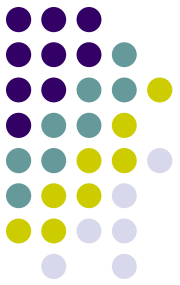


DFA for Numeric Literals

$(+|-|\lambda)(0|1|2|3|4|5|6|7|8|9)+|(.(0|1|2|3|4|5|6|7|8|9)+|\lambda)$

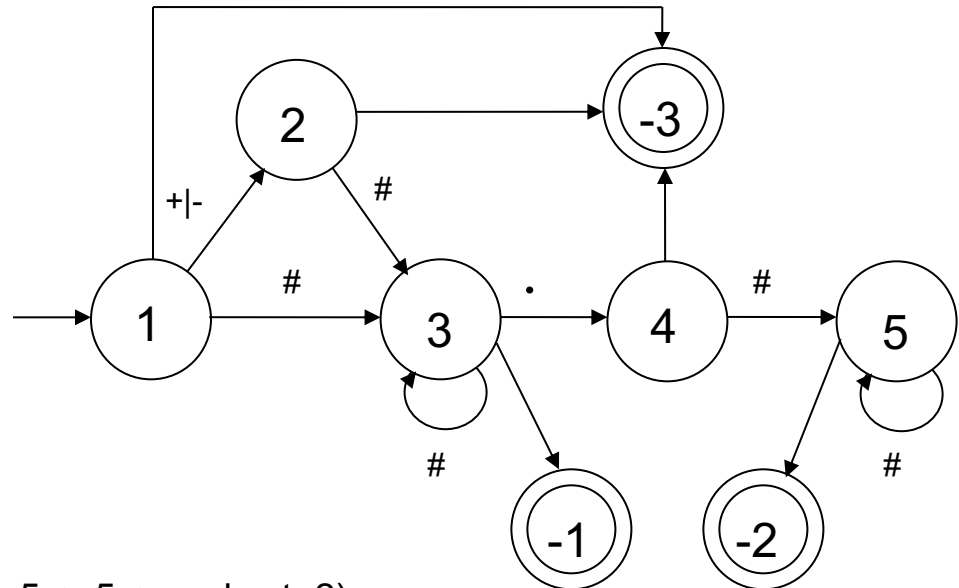
- a. 12
 - a. 1 -> 3 -> 3 OK!
- b. 1.2
 - a. 1 -> 3 -> 4 -> 5 OK!
- c. +12.34
 - a. 1 -> 2 -> 3 -> 3 -> 4 -> 5 -> 5 OK!
- d. 12.
 - a. 1 -> 3 -> 3 -> 4 -> ends No!
- e. .123
 - a. 1 -> ends No!
- f. 12.12.34
 - a. 1 -> 3 -> 3 -> 4 -> 5 -> 5 see . error

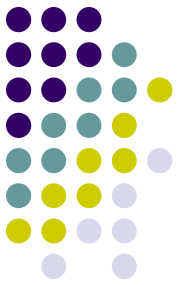




DFA for Numeric Literals – with terminating states (+|-|λ)(0|1|2|3|4|5|6|7|8|9)+|(.(0|1|2|3|4|5|6|7|8|9)+| λ)

- $_ \rightarrow$ represents a space
- 12 (1 \rightarrow 3 \rightarrow 3 OK!)
- 1.2 (1 \rightarrow 3 \rightarrow 4 \rightarrow 5 OK!)
- +12.34 (1 \rightarrow 2 \rightarrow 3 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 5)
- 12. (ends at 4)
- .123 (ends at 1)
- 12.12.34 (stops at 5 OK)
 - 12.12
- abcd (ends at -3)
- +abc (ends at -3)
- +_ (ends at -3)
- 4a (ends at -1)
 - 4
- 425_ (1 \rightarrow 3 \rightarrow 3 \rightarrow 3 \rightarrow ends at -1)
 - 4
- -12.345_ (1 \rightarrow 2 \rightarrow 3 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 5 \rightarrow 5 \rightarrow ends at -2)
 - -12.345
- What ends up at -1? integer
- What ends up at -2? double
- What ends up at -3? Non-numeric





Things to think about

- What does the g++ compiler do with
 - `int i = "Hello"; ?`
- What about
 - `char c = 'H'; ?`
 - `char c = 'H'; ?`
- What does the g++ compiler do with
 - `int i = -000; ?`
 - `cout << i << endl;`